Detecting Blackhole and Greyhole Attacks in Vehicular Delay Tolerant Networks

Yinghui Guo, Sebastian Schildt and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: [guo|schildt|wolf]@ibr.cs.tu-bs.de

***Abstract*—Blackhole and greyhole attacks can cause severe problems in Delay- and Disruption-Tolerant Networks (DTNs), where connectivity is intermittent and long delays are actually the norm. Traditional security protocols cannot completely address such problems in DTNs, hence an efficient algorithm to detect malicious nodes in DTNs is imperative. In this paper, we propose a misbehavior detection system to defend against blackhole and greyhole attacks. By collecting and securely exchanging data of previous encounters, a node can assess the trustworthiness of other nodes in order to detect blackhole and greyhole attacks. We evaluate our method through extensive simulations using different DTN routing protocols. Our simulation results show that even when the drop probability of greyhole attacks varies in a wide range, our approach can still efficiently detect evil nodes with a high detection rate and a low false positive rate while maintaining a low energy consumption.**

I. Introduction

In vehicular networks, many vehicular nodes participate in a dynamic wireless network and transfer messages between each other. However, due to the high movement of vehicular nodes, the connectivity in vehicular networks is highly unstable and links may change or break soon after they have been established. Delay- and Disruption-Tolerant Networks (DTNs) are designed to operate under such conditions. DTNs implement a "store, carry and forward" paradigm [4]. A packet will be sent over an existing link and buffered at a node until a connection to a suitable next hop is established. At least for the next years, until a high penetration of networked vehicles is realized, we believe that delay-tolerant methods are a necessity in such networks, leading to Vehicular Delay Tolerant Networks (VDTNs).

Common vehicular networks typically assume cooperation and no malicious behavior from participating vehicular nodes. However, vehicles are individual entities that can make independent decisions regarding the forwarding or deletion of messages. Some of the vehicular nodes may be malicious, trying their best to destroy or disrupt the network. Therefore, security considerations are clearly an important issue. Some work has focused on authentication and encryption [1]. Even though authentication and encryption are efficient methods to defend the system against outside attackers, it cannot safeguard the system from inside attackers, nor guarantee the willingness of nodes to cooperate. Hence, a flexible Misbehavior Detection System (MDS) is essential for VDTNs.

978-1-4673-5494-3/13/$31.00

Much work has been done in the area of MDS to detect or mitigate the effects of malicious nodes. In mobile ad hoc networks, the neighborhood monitoring approach is a traditional way to detect malicious nodes. Relying on individual nodes to monitor neighbors' traffic, the Watchdog MDS [11] and the reputation-based MDS called CONFIDANT [2] detect evil nodes and mitigate routing misbehaviors. However, due to the lack of long-lasting links in VDTNs, it is infeasible to monitor neighbors continuously.

Some recent works have suggested the usage of encounter tickets to tackle the problem of misbehavior detection in DTNs [8]. The idea is that after a contact and transmission of data between two nodes, they provide each other with a ticket about this encounter. When a node encounters another node, these tickets will be exchanged and used to classify their behavior and to detect blackhole attacks. A similar system is presented in [9]. Based on the contact records, a node can detect if other nodes have dropped packets. To prevent collusion of attackers, the contacted node needs to ask surrounding nodes for help. These surrounding nodes make decisions and send reports to other nodes. However, the MDS in [8] and [9] is designed to only detect blackhole attacks. Additionally, when attackers with the ability to forge encounter records exist, the normal nodes need to query surrounding nodes for information that will help them to correctly detect evil nodes. In our previous work [6] we implemented a MDS based on encounter records, in which nodes can detect and exclude blackholes from the network on their own judgement, without needing to vote on any decisions.

In this paper, we propose a general mechanism that not only detects blackhole but also greyhole attacks. Furthermore our system includes an incentive mechanism to encourage the cooperation of nodes. Depending on the amount of information available when two nodes meet, the system will adaptively choose a suitable detection threshold to maximize detection rates while minimizing false positives. Our MDS is designed to detect greyhole attacks with varying drop probabilities. We also introduce a trust reputation system to encourage cooperation. Messages from nodes with a good trust reputation will be accepted and forwarded by others, while nodes with a low trust reputation will be banished from the network.

The remainder of this paper is structured as follows: Section II introduces the vehicular node and attack model. In Section

III, we explain our system's architecture and detection scheme. The simulation-based evaluation is presented in Section IV. Finally, in Section V we draw our conclusions.

II. SYSTEM MODEL

A. Vehicular Node Model

We assume that each vehicular node possesses its own private and public key pair and an unique identifier. Furthermore we require the network to be loosely time synchronized: Vehicular nodes will be in the same time slot at any time. After successfully transmitting messages with another vehicular node, both vehicular nodes will generate an Encounter Record (ER). Additionally each vehicular node will store two lists in its memory: the Meeting List (ML) and the Local Blacklist (LBL).

1) *Encounter Record*: An ER will be generated after two vehicular nodes met and successfully exchanged messages. Here we use vehicular node i and vehicular node j as an example to illustrate how the ER is constructed. Vehicular node i generates the ER for vehicular node j as follows:

$$\begin{aligned} ER_i &= ID_i, ID_j, sn_i, sn_j, t, Re_{i\to j}, Re_{j\to i} \\ Re_{i\to j} &= \{(msg_{id}, msg_{src} | i \quad send \quad msg \quad to \quad j)\} \\ Re_{j\to i} &= \{(msg_{id}, msg_{src} | j \quad send \quad msg \quad to \quad i)\} \\ sig_i &= E_{RK_i}\{H(ER_i)\} \\ sig_j &= E_{RK_j}\{H(ER_i)\} \\ ER_i^* &= ER_i, sig_i, sig_j \end{aligned} \tag{1}$$

The ER includes both of the vehicular nodes' identifiers, ID_i and ID_j. Each vehicular node possesses its own unique sequence number *sn* that starts from 1 and is increased after each contact. A vehicular node is not allowed to use the same *sn* twice. t indicates the time when this ER was generated. The ERs in [9] use a vector of packets buffered by nodes, the identifiers of the packets received by nodes and the identifiers of the packets sent by nodes to detect blackhole attacks. This will make the size of the record too large if there is a lot of traffic in the system. In our system we introduce the Re set, which identifies the transmitted messages. The $Re_{i\to j}$ set consists of (id, src) 2-tuples storing for each message that has been received by j from i, the message's id and the id of the originating node. $Re_{j\to i}$ contains the message information sent from vehicular node j to i. Both communication partners need to cryptographically sign an encounter record (sig_i, sig_j) to ensure its authenticity and integrity. $E_{RK_i}\{*\}$ and $E_{RK_j}\{*\}$ denote the encryption using vehicular node i and j's private key. Here we use $H(*)$ to denote a hash function.

2) *Meeting List*: In this list, a vehicular node stores the information from previously encountered vehicular nodes. Each record in the ML includes the information of the identifier *ID* of the encountered vehicular node, the unique sequence number *sn* of the encountered vehicular node, the time t of the contact and the Trust Reputation (TR) (see section III-A) assigned to the encountered vehicular node. The ML entries can be used to check the validity of ERs later, as they store the last known (*sn*, *t*) combination for a vehicular node: In a normally operating network without any forged ERs, the condition holds that a greater *sn* for a given *ID* also implies a greater t. This relation will be used by the evaluation module (see section III-A) when checking new ERs.

3) *Local Blacklist*: All malicious nodes locally detected by a vehicular node will be put into that vehicular node's LBL. A vehicular node will refuse to transfer or receive messages from vehicular nodes in its LBL. To encourage more vehicular nodes to participate in the network, we provide an incentive mechanism for misbehaving vehicular nodes: All records in the LBL have an expiration time. After some predetermined time, an entry in the LBL is removed and the previously detected node is allowed back into the network. However, if a node from the LBL is allowed back to the network, it is on "probation", meaning it will start with a lower initial TR than other nodes that have just joined the network. If a node is removed from the LBL and assigned a new initial TR, the current entries for that node in the ML will not be changed.

B. Attack Model

In VDTNs, a vehicular node may agree to forward packets but actually drop them, because it is malicious. The most common attacks are blackhole and greyhole attacks. Blackhole attackers may advertise many excellent routes through themselves and then drop all packets. When only partial droppings occur, the attack is referred to as greyhole attack, which is much more difficult to detect [5]. Considering energy and memory, without giving incentives, normal nodes may be reluctant to cooperate if it is not directly beneficial to them. Therefore, our MDS will focus on dealing with blackhole and greyhole attacks, while encouraging normal nodes to forward other nodes' messages. To conceal the packet dropping, we assume malicious nodes will use the following behaviors to deceive other vehicular nodes:

1) *Behavior 1*: A malicious node will first receive messages, and then according to its drop probability, the malicious node decides whether it drops the message. If a given node's drop probability is 1, it is a blackhole. Drop probabilities > 0 and < 1.0 characterize a greyhole attack.

2) *Behavior 2*: It is not practical to store all ERs, hence in our system a normal vehicular node only stores w new and sequential ERs in its memory to provide them to other vehicular nodes for verification. However, a malicious node will store as many beneficial ERs for itself as it can and randomly choose these better ERs from its memory to present them to others.

3) *Behavior 3*: To get good ERs and maintain the sequence of its *sn*, a malicious vehicular node can use the same *sn* to generate ERs with different vehicular nodes until the content of the ER is beneficial for it, afterwards it updates the *sn* to generate new ERs.

4) *Behavior 4*: A malicious vehicular node generates a group of sequential and good ERs by adhering to the rules. After it accumulates enough good ERs it commences the attack, but does not store any new ERs generated during the attack phase. After a while it may decide to generate a new set of sequential,

大的sn值和新的t值相互对应，(sn,t)以此可以断定出ER的伪造//

1.检查者博弈 可不可以用？
2.更小的TR对我们来讲[0,1]本来是0.5现在是更接近0的值//

一旦加入黑名单(LBL)就不收也不发，expiration time超时时间，允许LBL名单上的节点返回到网络中

报文的丢弃行为 可能通过以下手段来隐藏。

1.丢弃概率-grayhole方式

2.正常节点存储最新的w个证据，恶意节点存储有利的证据（部分证据）

3.用同样的sn 放在ER里

4.执行一会儿好行为 积累ER; 然后坏行为 不记录ER; 如此反复。

w个ER;
每个ER里都保存了 本次encounter中j节点

good ERs by abiding the rules of the network.

III. System Architecture

The presented MDS system can be decomposed into two main components: the evaluation module and the decision module.

A. Evaluation module

In the evaluation module, vehicular nodes assesses the trustworthiness of other vehicular nodes. The range of the TR is between 0 and 1. When a vehicular node first joins the network, it assumes an initial TR of 0.5 for vehicular nodes it meets for the first time. If a vehicle added another vehicular node to its LBL in the past, but that entry has just expired, it will assume a new initial TR of 0.4 for the previously blacklisted node. The TR will be updated by the evaluation module. When vehicular node i encounters j, the evaluation module will check the following conditions:

TR范围[0,1]
如果node第一次加入，初始化为0.5;
如果曾经进入LBL黑名单，则初始化为0.4

1) If vehicular node j is in vehicular node i's LBL, vehicular node i will refuse to transfer and receive messages from vehicular node j and the following checks will be omitted.

2) When two vehicular nodes meet, both need to provide up to w new and sequential ERs to each other. If vehicular node j behaves well, the sn in its w ERs should be sequential. If vehicular node i find the sn of vehicular node j is not sequential, vehicular node i will add j to its LBL.

j提供w个最新的ER

*判断sn值是否连续的
*sn和t都是新的大于旧的

也就是 本地ER伪造问题

3) If vehicular node i met j before, it checks the consistency of j's sn and t fields in the reported ERs: Compared to the information in the ML, for every sn that is larger than the sn for j in the ML it must hold, that the time t from the ER is also larger than the t recorded in the ML for vehicular node j. If an inconsistency is detected, vehicular node i will add j to its LBL.

4) In this step j's ERs are used to update i's ML. Vehicular node i checks sn and t of j's encountered vehicular nodes. If i never met a node mentioned in ERs, it will add the node's *ID* and appropriate sn and t values to its ML. If j's ERs contain newer sn and t records for a node already in the ML, the corresponding ML entries will be updated. However, if vehicular node i finds that there are contradictions between sn and t of j's encountered vehicular nodes with sn and t of the same vehicular nodes in its ML, these vehicular nodes encountered by j will be added to vehicular node i's LBL, and the corresponding ERs will not be used for the following checks. Integrating information from another node's ERs into their ML allows nodes get information about nodes that they never met by themselves. This means updated knowledge is propagated quickly through the network.

从j的ER<比如和a相关的ER>里，
i知道了 j和a的sn 和t

i应该把这个信息保存下来
如果存在矛盾a的id将被加入i的黑名单

也即是把最新的sn和t快速传播开来

5) If vehicular node j passes the checks 1 to 4, vehicular node i will use the $Re_{i\to j}$ and $Re_{j\to i}$ sets to update the TR of j. Vehicular node i first figures out the message forwarding ratio of j, which is the total number of messages that are sent out by j in w ERs over the total number of messages received by j. Vehicular node j provides w ERs: ER_0 , ER_1,.., ER_{w-1}. $N_{send}^{ER_0}$, $N_{send}^{ER_1}$, …, $N_{send}^{ER_{w-1}}$ denote how many messages are sent out by j in ER 0, 1, … w-1. $N_{recv}^{ER_0}$, $N_{recv}^{ER_1}$, …, $N_{recv}^{ER_{w-1}}$ denote how many messages are received by j in ER 0, 1, … w-1. The message forwarding percentage θ can be expressed as formula (2).

$$\theta = \frac{\sum_{m=0}^{m<w} N_{send}^{ER_m}}{\sum_{m=0}^{m<w} N_{recv}^{ER_m}} \tag{2}$$

j在这w个ER中
θ =总send数/总recv数

θ 有阈值Nth

若 θ 小于阈值，则不合格TR减少（本地评价）
TR=TR- γ

若 θ 大于阈值，则合格进入下一步

If $\theta > N_{threshold}$, we proceed to step 6. $N_{threshold}$ is determined dynamically depending on the amount of ERs available. For details see section III-C. If $\theta < N_{threshold}$ this indicates that vehicular node j may selectively drop messages. The TR of j will be decreased according to formula (3).

$$TR_i^j = TR_i^j - \gamma \qquad (0 < \gamma < 1) \tag{3}$$

6) The blackhole and greyhole attackers drop messages, so they prefer to transmit the messages which are generated by themselves. To detect this, we use another mechanism to find malicious vehicular nodes. Again $N_{send}^{ER_0}$, $N_{send}^{ER_1}$, …, $N_{send}^{ER_{w-1}}$ denote how many messages are sent out by j in ER 0, 1, … w-1. $N_{send}^{jER_0}$, $N_{send}^{jER_1}$, …, $N_{send}^{jER_{w-1}}$ are the number of messages which j generated by itself and sent out in ER 0, 1, … w-1. The vehicular node's own message forwarding percentage ψ is defined as follows:

$$\psi = \frac{\sum_{m=0}^{m<w} N_{send}^{jER_m}}{\sum_{m=0}^{m<w} N_{send}^{ER_m}} \tag{4}$$

ψ =j自己生成的报文send数/总send数

ψ有阈值NRth

若ψ大于阈值，则 TR=TR- γ ；若这两步同时作为坏行为(Nth NRth)
TR=TR- ρ

若这两步都没检测出
TR=TR+ λ

$\psi \geq NR_{threshold}$ indicates that vehicular node j prefers to send its own messages. To punish this kind of behavior, we use formula (3) to decrease the TR. If step 5 and 6 simultaneously detect bad behaviors of vehicular node j, instead of using formula (3) twice, we use formula (5) to decrease the TR.

$$TR_i^j = TR_i^j - \rho \qquad (\gamma < \rho < 1) \tag{5}$$

If neither of these steps detects abnormal behaviors, this indicates that vehicular node j is normal and its behavior will be encouraged by increasing its TR according to formula (6).

$$TR_i^j = TR_i^j + \lambda \qquad (\gamma < \lambda < 1) \tag{6}$$

To encourage more vehicular nodes to participate in the network, the additive component λ is larger than the subtractive component γ. Only when vehicular node i is convinced that vehicular node j behaves badly because both thresholds from step 5 and 6 are triggered, the larger subtractive component ρ is applied to decrease the TR of j. Reasonable values for these parameters can be chosen empirically. In this paper we choose γ to be 0.04, ρ to be 0.09 and λ to be 0.06 for Epidemic [13], MaxProp [3] and PROPHET [10] routing. For Spray and Wait routing [12] we set γ to 0.02, ρ to 0.05, λ to 0.03 and $NR_{threshold}$ to 0.7.

After step 4, only the eligible ERs are left to be used in the calculation of step 5 and 6. If the number of the eligible ERs is < 10, the maximal decrease of TR is limited to 0.02 after step 5 and 6, preventing mistakes from making decisions based on insufficient information. The updated TR will be delivered to the decision module.

有效ER少的话【仿真刚刚开始】，最大下降值为0.02

B. Decision Module

The decision module is responsible for making an appropriate decision after vehicular nodes receive an updated TR. If the updated TR_i^j is less than the evil threshold T_{evil} (0.3), vehicular node i adds j to its LBL and refuses to exchange messages with j. If the updated TR_i^j is more than the friend threshold T_{friend} (0.8), it indicates vehicular node j prefers to forward messages for others, therefore vehicular node i may trust vehicular node j and list it as friend. When vehicular node i simultaneously communicates with multiple vehicular nodes, it will first transfer messages with its friend node j, then forward messages to normal nodes. Otherwise, vehicular node i defines vehicular node j to be normal and transfers messages to j. Besides, vehicular node i will update the information of vehicular node j in its ML.

C. Adaptive Threshold

Upon connecting, two vehicular nodes first exchange their ERs prior to exchanging any application data. The ERs provided by a communication partner are used to assess its trustworthiness. Depending on the number of ERs provided by the communication partner a node chooses an adaptive threshold ($N_{threshold}$) which is applied in step 5 of the evaluation module. The rationale is that a larger number of ERs contains more information and thus the threshold can be stricter because the risk to misjudge a node is smaller. However, in VDTNs the connectivity is intermittent hence the system cannot guarantee that every vehicular node has enough ERs. Therefore, in case a node cannot provide enough ERs, a more relaxed threshold is applied. This ensures that the system will not generate too much false positives while still maintaining the ability to detect obvious offenders.

如果ER个数少，Nth(第5步)应该更小(宽松一些)为了应对误报

ER数目上升时，好节点的θ上升；坏节点的θ下降

最优的Nth应该与Nth相关

Nth = b-a/w

Appropriate functions determining suitable $N_{threshold}$ values have been found using extensive simulations (see section IV-A). The results show that under the same drop probability, the normal nodes' message forwarding ratio θ increases as the vehicular nodes store more ERs in their memory. Meanwhile, the malicious nodes' message forwarding ratio decreases when the vehicular nodes provide more ERs under the same drop probability. We obtained optimal $N_{threshold}$ values for different drop probabilities and number of ERs using extensive simulations. To achieve a high detection rate but also maintain a low false positive rate, we choose the minimal $N_{threshold}$ of different drop probabilities using the same number of ERs as the final $N_{threshold}$. The results indicate that the optimal $N_{threshold}$ is highly correlated to the number of available ERs and can be approximated by formula (7):

$$N_{threshold}(w) = b - \frac{a}{w} \quad (a > 0,\ b > 0,\ w\text{: number of ERs}) \tag{7}$$

Figure 1 shows the optimal threshold functions for different routing protocols. As routing protocols differ in the way they forward messages, they lead to different "normal" values for the ratio used in step 5 of the evaluation module and thus different threshold functions are needed to guarantee optimal system performance under different routing protocols. Usually the routing protocol used in a system is fixed, therefore routing dependent threshold functions are not a problem when deploying our MDS. Figure 1 shows, that if the number of ERs is above 80, the detection threshold is nearly stable. Considering the memory of vehicular nodes, we define the number of ERs (w) in nodes' memory will be between 10 and 100. Nodes should store as much ERs as possible up to 100 as evidence for their cooperating behavior.

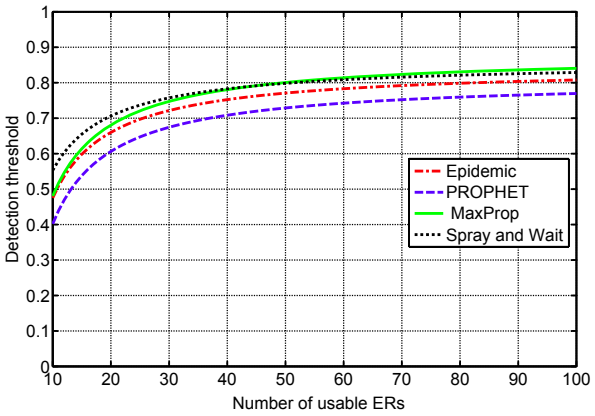


Figure 1. Variable Detection Threshold

IV. Performance Evaluations

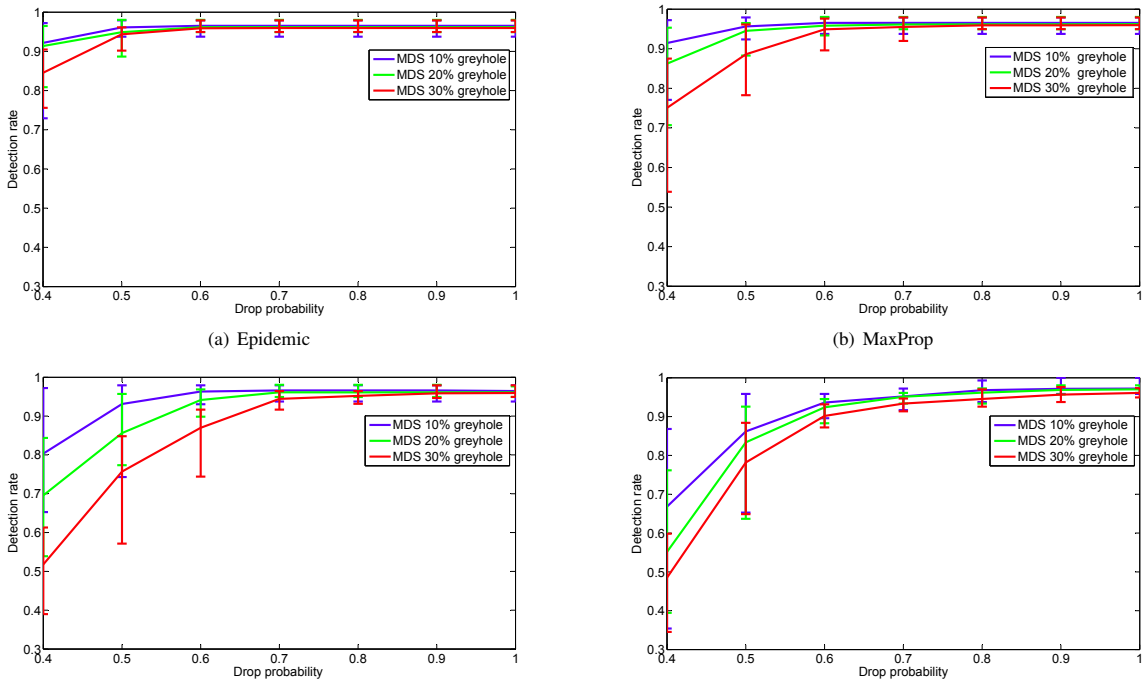
A. Simulation Setup

We use The ONE simulator [7] to evaluate our MDS. In our simulations 40 vehicular nodes with a transmission radius of 100 meters and a moving speed varying from 10 km/h to 50 km/h and a buffer size of 10 MB are uniformly deployed in the Helsinki city map with a size of 4500 m × 3400 m to simulate a VDTN. Vehicular nodes use the shortest path map based movement model, generate messages in an interval between 25 and 30 seconds. We performed comparative measurements using the Epidemic, MaxProp, PROPHET and Spray and Wait routing protocols. We randomly choose 4, 8 and 12 nodes among the 40 nodes as malicious nodes whose drop probability varies from 0.4 to 1 and these malicious nodes will independently attack the system. The simulation time is 12 hours (43200s). The simulation results presented for each scenario are the average, min and max results of 10 experimental runs. Vehicular nodes apply the MDS starting from second 10000.

We use the following metrics to evaluate our MDS.

1) *Detection rate*: The percentage of evil nodes that have been detected by all good nodes. For a detection rate of 100%, we require that *all* evil nodes are detected by *all* normal nodes. Thus in our system the detection rate is defined as:

$$d\_rate = \frac{\#true\_pos.}{\#normal\ nodes \times \#malicious\ nodes} \tag{8}$$



(a) Epidemic

(b) MaxProp

(c) PROPHET

(d) Spray and Wait

Figure 2. Detection Rate

2) *False positive rate*: The percentage of good nodes that are mistakenly detected as evil nodes:

$$fp\_rate = \frac{\#false\_pos.}{\#false\_pos. + \#true\_neg.} \quad (9)$$

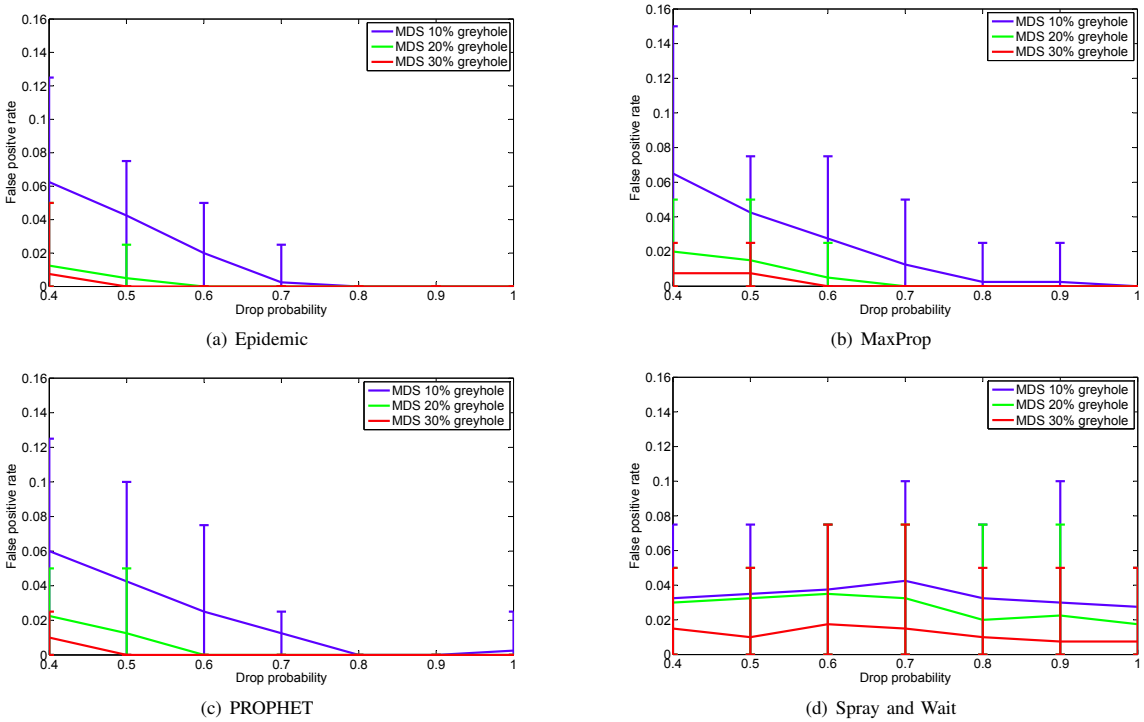
3) *Energy consumption*: The total number of relayed messages.

B. Detection Rate

We studied the detection rates versus different number of malicious nodes using different drop probabilities under four different routing protocols. To better understand the performance of the MDS the average and the maximum and minimum detection rates from 10 runs are shown in Figure 2. Any evil node should be purged from the network as fast as possible. However, malicious nodes with a high drop probability will have a more severe impact on the network performance, therefore they should be detected quickly. As seen in Figure 2, dealing with malicious nodes with a high drop probability, our system performs well and can achieve a high detection rate up to 97%. For the malicious nodes with a low drop probability that have a less pronounced effect on the network performance, our MDS needs to take a longer time to detect all of them. Under the current simulation time, the detection rate of malicious nodes with a lower drop probability is not as high as the detection rate of malicious nodes with a higher drop probability. However, when the simulation time is long enough, our MDS will also achieve a high detection rate for malicious nodes with a low drop probability.

C. False Positive Rate

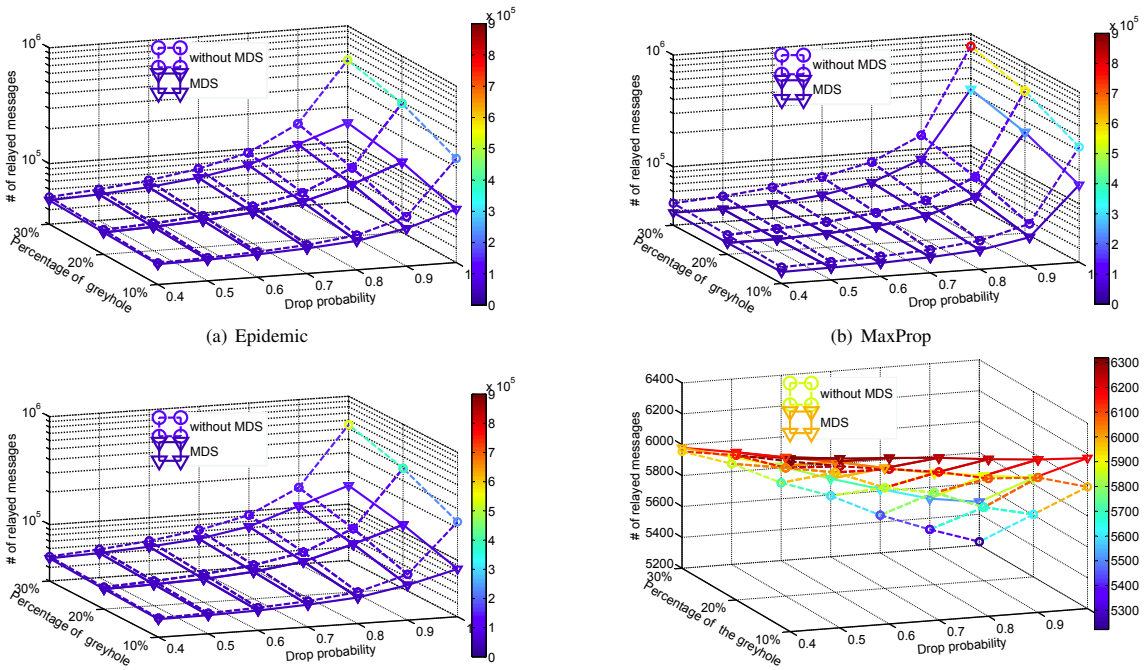
Figure 3 presents the false positive rates versus the percentage of malicious nodes under four different routing protocols. It can be seen, that our MDS can achieve a low false positive rate using different routing protocols: When the drop probability is high, there is a distinctive difference between the behavior of normal vs. malicious nodes. With a drop probability of 1 (blackhole attacks), our system achieves zero false positive under routing Epidemic and MaxProp. When the drop probability is low, the behaviors of malicious nodes are very similar to the behaviors of normal nodes, therefore it gets harder for the algorithm to discriminate between good and bad nodes and thus the false positive rate is larger. When the drop probability is 0.4, the average false positive rate of our system is around 6% in different scenarios. But when the drop probability exceeds 0.5, the average false positive rate of our system is less than 5% in different scenarios. As shown in Figure 3, as the percentage of malicious nodes in the system increases, the average false positive will decrease. The results depicted in Figure 3 also show, that as the number of evil nodes increases, nodes have more opportunities to meet malicious nodes and thus can make a more precise decision.



(a) Epidemic (b) MaxProp

(c) PROPHET (d) Spray and Wait

Figure 3. False Positive Rate



(a) Epidemic (b) MaxProp

(c) PROPHET (d) Spray and Wait

Figure 4. Energy Consumption

D. Energy Consumption

The energy consumption of a network is closely correlated to the number of transferred messages, as each message needs a certain amount of energy to be sent, received and processed. Figure 4 presents the average number of messages relayed in the system versus the percentage of malicious nodes using different drop probabilities in the system under four different routing protocols. The x axis shows the drop probability. The y axis shows the percentage of evil nodes in the system. The z axis shows the total number of relayed messages in the system. Figure 4(a) to 4(c) show that, especially for routing protocols with unlimited replication such as Epidemic, MaxProp or PROPHET, greyhole attacks can cause a drastic increase in relayed messages. Independent of the drop probability or the percentage of evil nodes in the system, our MDS always maintains a lower number of relayed messages comparing to the number of relayed messages without the MDS. Hence, by detecting and excluding evil nodes from the network, our MDS saves large amounts of energy for the system. When there are 30% of blackhole attackers in the system, our system achieves energy savings of 71%, 59%, 73% for the Epidemic, MaxProp and PROPHET routing protocols respectively.

Routing protocols which limit the number of replicas such as Spray and Wait do not suffer so much from increased relaying. In our setup each message is allowed to be copied 6 times by the Spray and Wait routing, hence the total number of relayed messages has an upper bound. When the malicious nodes drop the messages, the number of relayed messages will be decreased. As our MDS decreases the chance that the limited replicas are relayed to the evil nodes, the total number of relayed messages by using our MDS is closed to the upper bound. For routing protocols with a limited number of replicas this is desired, as it increases the probability that messages can be forwarded towards their destinations and thus increases the delivery rate in the system.

V. Conclusion

We presented a MDS that enables nodes in a VDTN to independently detect malicious nodes by distributing and combining information from previous encounters in the network. The system excludes malicious nodes from the network, and thus prevents them to further disrupt the network. The integrated reputation system encourages selfish nodes to cooperate. Our extensive simulations under different routing protocols demonstrate that our MDS can achieve a high detection rate and a low false positive rate for different scenarios where the number of malicious nodes, the attack intensity or the employed routing protocol is varied. Finally, our MDS can significantly reduce energy consumption for common DTN routing protocols, which create a large number of replicas.

References

[1] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo. Applicability of identity-based cryptography for disruption-tolerant networking. In *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, MobiOpp '07, pages 52–56, New York, USA, Jun. 2007.

[2] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '02, pages 226–236, Lausanne, Switzerland, Jun. 2002.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-tolerant networking architecture. In *RFC 4838 (Informational)*, Apr. 2007.

[5] M. Chuah and P. Yang. Comparison of two intrusion detection schemes for sparsely connected ad hoc networks. In *Military Communications Conference 2006*, pages 1–7, Washington DC, Oct. 2006.

[6] Y. Guo, S. Schildt, J. Morgenroth, and L. Wolf. A misbehavior detection system for vehicular delay tolerant networks. In *Proceedings of the INFORMATIK 2012*, Braunschweig, Germany, Sep. 2012.

[7] A. Keränen, J. Ott, and T. Kärkkäinen. The one simulator for dtn protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Simutools '09, pages 55:1–55:10, Rome, Italy, Mar. 2009.

[8] F. Li, J. Wu, and A. Srinivasan. Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets. In *INFOCOM 2009, IEEE*, pages 2428–2436, Rio de Janeiro, Brazil, Apr. 2009.

[9] Q. Li and G. Cao. Mitigating routing misbehavior in disruption tolerant networks. *Information Forensics and Security, IEEE Transactions on*, 7(2):664–675, Apr. 2012.

[10] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, Jul. 2003.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 255–265, Boston, USA, Aug. 2000.

[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 252–259, Philadelphia, USA, Aug. 2005.

[13] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, 2000.